International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 16, No.1 (Aug-24)

http://dx.doi.org/10.12785/ijcds/160146

Rapid Navigation Optimization - based Deep Convolutional Neural Network for Covid-19 Detection using CT Scans

Priya Sawant[1*] **and R Sreemathy**[1]

[1]*SCTR's Pune Institute of Computer Technology, Affiliated to Savitribai Phule Pune University, Pune, India*
Received 1 Nov. 2023, Revised 19 Apr. 2024, Accepted 26 Apr. 2024, Published 1 Aug. 2024

Abstract: In December 2019 a highly infectious virus named 'Severe Acute Respiratory Syndrome Coronavirus 2 (SARS-COV-2) sparked a global pandemic. Deep Neural Networks have been extensively used to develop intelligent systems for accurate and timely diagnosis of COVID-19 infection using chest Computerized Tomography. However, Deep Learning approaches require a large annotated dataset. The fundamental goal of this research is to develop a model that would learn efficiently from a size-limited dataset. This study proposes a hybrid feature extraction approach. Our proposed technique exploits the CT imaging characteristics of COVID-19 infection through hand-crafted texture features and complex features extracted by a pre-trained ResNet101 network. The 7-layered Deep Convolutional Neural Network used for classification is optimized using a revolutionary rapid navigation optimization technique. The proposed optimization improves the Moth-Flame Optimizer by integrating the concept of Mayfly velocity to update the position of the Moth fly in the exploration space. When tested on an open access dataset, COVID - CT containing 349 COVID-19 positive CT images and 397 COVID-19 negative CT images, the accuracy, sensitivity, and specificity of the proposed rapid navigation optimization-based deep CNN classifier were 97.260%, 94.301%, and 99%, respectively. The proposed model was also tested on an augmented COVID - CT dataset and a larger dataset, COVIDx-CT-3A. The proposed model has exhibited an accuracy of 99.61% and 89.35% on the augmented COVID - CT dataset and COVIDx-CT-3A, respectively. Our proposed method outperformed the other published cutting-edge research works that have tested on the small COVID - CT dataset.

Keywords: COVID-19, chest CT, Haralick texture features, Local Directional Pattern, Gray level co-occurrence matrix, DNN, Moth-Flame Optimization

1. Introduction

In December 2019 the world was hit by a major pandemic caused by a novel coronavirus called 'Severe Acute Respiratory Syndrome Coronavirus2 (SARS-COV-2). As COVID-19 is highly infectious, early diagnosis and subsequent isolation of the infected patient is imperative to control its rapid spread and decide the future course of treatment. A patient exhibiting typical COVID-19 symptoms like fatigue, sore throat, shortness of breath, diarrhea, headache, high fever, and loss of taste is prescribed to undergo the Reverse Transcription-Polymerase Chain Reaction (RT-PCR) test for further diagnosis [1]. However, RT-PCR test is reported to exhibit high false negative rates, particularly at the early stage of the disease. Further, shortage of testing kits in a pandemic scenario and a highly controlled testing environment hampers rapid and accurate diagnosis in case of infected patients. Chest Computerized Tomography (Chest CT) has played an important supplementary role in diagnosis, and follow-up assessment to determine degree of pulmonary involvement. Chest CT screening offers good resolution and a 3-D view of the lung. Another advantage of chest CT scans is the unique relationship between CT density and lung air content. Thorax of a normal healthy person consists of a large area of air and a few soft tissues like trachea and aorta. Air, with a Hounsfield Unit (HU) of 1000 appears as all black, while the other soft tissues exhibit patterns as per their HU number. COVID-19 infected patients exhibit Ground Glass Opacity (GGO), linear opacification, crazy-paving sign, and pulmonary consolidation [2] and [3]. Fig. 1 illustrates CT images of a normal lung (with Superior vena cava, Trachea and Aortic arch) and various CT imaging characteristics of COVID-19 infection [4].

Abnormalities that manifest in the radiographic images can be interpreted only by expert radiologists. In the event of a pandemic and scarcity of trained radiologists, accurate diagnosis and subsequent treatment is a challenge for healthcare practitioners. Computer Aided Diagnosis (CAD) can assist the clinical diagnosis process and increase the rate of early diagnosis with high accuracy [5]. Researchers have proposed many machine learning - based techniques

E-mail address: jopent8@gmail.com, rsreemathy@pict.edu

for detection of COVID-19 infection.

In this research, the authors propose to exploit texture characteristics of chest CT images to develop a classifier to distinguish between COVID-19 and non-Covid-19 images. To enhance the classifier performance, ReliefF-based feature selection algorithm and a novel nature-inspired optimizer are incorporated in the process of classification. The major contributions involved in this research are as follows,

Hybrid feature extraction: As COVID-19 lesions affect pixel intensity values in CT images, we exploit Haralick texture features to identify abnormalities. We compute a joint feature descriptor based on Local Binary Pattern and Local Directional Pattern codes to integrate texture, statistical and gradient features. To extract more complex and subtle features from the computed texture features we use a pre-trained ResNet101.

ReliefF-based feature selection: The most relevant features are selected using ReliefF algorithm. ReliefF algorithm considers 'k' neighboring 'hits' and 'misses' to estimate the feature quality and relevance. The algorithm effectively eliminates the redundant features and improves classification accuracy.

Novel Rapid Navigation Optimization-based deep CNN: Tuning of many parameters in Deep Neural Networks (DNNs) is a critical issue. A revolutionary Rapid Navigation Optimization (RNO) technique is implemented to enhance the Moth-Flame Optimization (MFO) algorithm. The velocity concept of Mayfly algorithm is integrated in MFO to update position of Moth fly in the exploration space. The Rapid Navigation Optimizer (RNO) is a hybrid of two nature-inspired optimization algorithms, the Mayfly Algorithm (MA) and the Moth-Flame Optimizer (MFO). The proposed optimizer updates weights in the DCNN and improves classification performance on a small dataset.

The rest of this paper is structured as follows: A summary of related research work carried out to detect COVID-19 from chest CT images is presented in Section 2. Section 3 describes the proposed methodology. Comparative performance analysis of the proposed RNO is presented in Section 4. Finally, Section 5 concludes the paper. Additionally, it discusses limitations and future research directions.

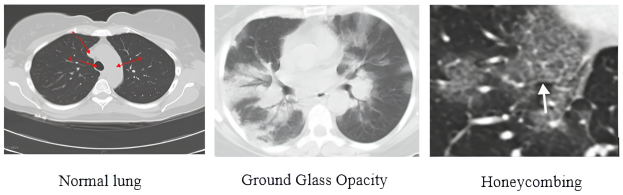

Figure 1. Radiological patterns on Chest CT scans

2. Related Work

Researchers have applied Digital Image Processing (DIP) and Artificial Intelligence (AI) concepts to develop CAD systems to detect COVID-19 infection using Chest CT scans [6]. Hossain M. M. et al. [7] proposed an automatic detection model for COVID-19 diagnosis from chest Computed Tomography (CT) scans. A feature vector was constructed through the fusion of features extracted from two Convolutional Neural Network (CNN) models, VGG-19 and ResNet-50. Feature optimization methods, Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Linear Discriminant Analysis (LDA) were used to identify the most relevant features. The optimized features are finally classified with the Max Voting Ensemble Classification (MVEC). The fused features of VGG-19 and ResNet-50, processed with PCA and MVEC, exhibited highest values of accuracy, specificity, sensitivity, and precision at 98.51%, 97.58%, 99.49%, and 97.47%, respectively, after 5-fold cross-validation. Hassan E. et al. [8] evaluated pre-trained transfer learning models such as ResNet-50, VGG-19, VGG-16, and Inception V3 for classifying the input CT images. This study employed binary cross-entropy metric to compare COVID-19 cases with normal cases. Overfitting issues were addressed by use of the Stochastic Gradient Descent and Adam optimizers. The proposed pre-trained transfer learning models achieved accuracies of 99.07%, 98.70%, 98.55%, and 96.23%, respectively. Punitha S. et al. [9] proposed an Ant Bee Colony optimized ANN (ABCNN) to detect COVID-19 from lung CT scans. Nature-inspired optimizers were used for feature extraction, feature selection and optimization of the ANN. They achieved an accuracy of 92.37% on the public database compiled by Yang et al. [10]. Gao, He and Li [11] proposed a Swin-Unet- based light weight ANN for semantic segmentation of COVID-19 lesion in Chest CT scans. On a large public dataset of COVID-19 CT scans, they achieved precision, recall, and Intersection over Union (IoU) values of 0.780, 0.848 and 0.683, respectively. Model's accuracy depends on a large annotated dataset and may mis-predict if there is a slight variation or similarity in COVID-19 lesions and normal lungs. Tuncer I. et al. [12] combined swin-based patch division with textural feature extraction and developed the swin-textural model for automated diagnosis of COVID-19 on chest CT images. The swin-textural model had an accuracy of 98.71% on a data set containing total 4171 CT images from 210 subjects. Li J. et al. [13] proposed a hybrid model to improve COVID-19 classification accuracy of deep learning model using CT scans. Their proposed model combined a semi-supervised domain adaption model and an Extreme Learning Machine (ELM) classifier. A novel multi kernel correntropy induced loss function in transfer learning is introduced. Their model achieved 97.63% accuracy, 95.16% precision, and recall of 95.32%. Sadi M. S. et al. [14] proposed COV-CTX, a deep learning – based model to detect COVID-19 from CT-scan and X-ray images. Their system comprises three CNN models, VGG16, VGG16-InceptionV3-ResNet50, and Francois CNN. A large number

of CT-scan and X-ray images were individually used to train the models. Finally, the results of the models are combined and input to a voting ensemble of classifiers. The proposed system, COV-CTX attains 92.68% Cohens Kappa score for CT-scan image based COVID-19 detection. Farjana A. et al. [15] addressed the concern of limited access to large datasets of CT scan images due to privacy concerns in their study. They employed transfer-learning pre-trained models to automatically detect COVID-19 cases from CT scan images. The proposed methodology utilized VGG19, RESNet50, and DenseNet169 architectures. Experimental findings indicate DenseNet169 performed the best with an accuracy of 98.5%. Mohbey K. K. et al. [16] presented a DL technique and evaluated it on a dataset with 5000 images to identify COVID-19. Their transfer learning model is based on VGG-19 architecture. It attained an accuracy of 95% with minimum time penalty. Islam et al. [17] proposed a novel Convolutional Neural Network (CNN) to extract 100 prominent features from 2482 chest CT scans. The extracted features were deployed to several machine learning algorithms, viz, Gaussian Naïve Bayes (GNB), Support Vector Machine (SVM), Decision Tree (DT), Logistic Regression (LR) and Random Forest (RF). In the last stage, a voting ensemble – based model was implemented for COVID-19 detection. They achieved highest accuracy of 99.73%. A limitation of this study is that they implemented only the existing machine learning algorithms. Ren Q. et al.[18] proposed a hybrid framework, the Complex Shearlet Shattering Transform – Wide Residual Network (CSST-WR2N) to extract more granular multiscale discriminative features from CT images for COVID-19 detection. The model was trained on total 1599 COVID-19 CT images and 1627 non-COVID-19 CT images. The hybrid model achieved accuracy of 95.38%. However, the use of fixed Shearlet transforms may limit its performance. DeepPneumonia, a pre-trained ResNet-50-based system was developed by Ying Song et al. [19] to detect COVID-19 and localize the infection lesions. The hospital image dataset included COVID-19 CT images of 88 patients and CT scans of 101 Community Acquired Pneumonia (CAP) patients. They achieved 94% accuracy. The model could predict well on the original hospital dataset but could not predict external data directly. Li Caizi et al. [20] developed a self-ensembled co-training framework, trained by a limited labeled dataset of 33 COVID-19 CT images from hospitals and 70 unlabeled CT images. Values of performance metrics achieved by their semi-supervised methodology are: AUC 89.57%, sensitivity 79.38%, and specificity 99.75%. There is a possibility of unlabeled data being wrongly predicted due to lack of prior knowledge in the semi-supervised method. Balaha H. M. et al. [21] designed a hybrid learning and optimization model, the CovH2SD for detecting COVID-19 from the Chest Computed Tomography (CT). CovH2SD employed deep learning and pre-trained models to extract the features from the CT scans. The use of Harris Hawks Optimization (HHO) algorithm enhanced its diagnostic capabilities. CovH2SD achieved top accuracy of 99.33% on a large dataset comprising of total 15,535 CT images. Kaur T. et al. [22] proposed a model based on the deep features and Parameter Free BAT (PF-BAT) - optimized Fuzzy K-nearest neighbor (PF-FKNN) to detect COVID-19. The fully connected layer of transfer learned MobileNetv2 was employed to extract features. The features were classified by a FKNN classifier The proposed model achieved a validation accuracy of 99.38% on a large dataset comprising of 1252 CT scans of COVID-19 positive patients and 1230 COVID-19 negative images. Sen S. et al. [23] proposed a bi-stage feature selection approach comprising two filter methods, Mutual Information (MI) and ReliefF in the first stage and a bio-inspired optimizer, the Dragon fly optimizer in the second stage. The model exhibited a higher prediction rate of 98.39% on a large size dataset (1252 images of COVID-19 and 1230 images of non-infected patients) and accuracy of 90.00% on the smaller public dataset [10]. They attribute the misclassified cases to poor quality of certain images and lack of ample historical COVID-19 data. Wang Bo et al. [24] proposed a two-stage deep neural network - based model to classify between COVID - 19 positive and negative scans and highlight the lesion regions as well. The model used several segmentation models such as Fully Convolutional Networks (FCN), U-Net, V-Net and 3D U-Net++. In the classification stage, classification models such as ResNet-50, InceptionNet, Dual Path Network-92 (DPN-92) and Attention ResNet-50 were evaluated. The hospital acquired data set comprised 1136 training cases from 5 hospitals with 723 COVID-19 positive cases and 413 cases of variety of pulmonary diseases. The model achieved sensitivity of 0.974 and 0.922 specificity. However, the model relies heavily on a large annotated dataset. Yasar and Ceylan [25] presented a comparative study of techniques used to detect COVID-19. Grey Level Co-occurrence Matrix (GLCM) and Local Binary Pattern (LBP) methods were used to extract texture features which were subsequently fed to a 23-layered CNN classifier. On a dataset comprising 386 chest CT images of COVID-19 patients, the model obtained 95% accuracy, 94% sensitivity and 99% specificity with 70% training data. Mishra, Das et al. [26] proposed a decision fusion-based approach to identify COVID-19 from chest CT images. The model evaluated the fusion of several Deep CNN models, viz. VGG16, Inception V3, ResNet50, DenseNet121 and DenseNet201 on a publicly available dataset [10]. The fusion model achieved higher accuracy (88.34%) than the individual models. Shaban W. M. et al. [27] computed a set of features from chest CT scans using Grey Level Co-occurrence Matrix (GLCM). Selected relevant features were inputted to an Enhanced K-Nearest Neighbor (EKNN) classifier to classify between COVID-19 and non-COVID images. They achieved an accuracy of 96% and 71% sensitivity with minimum time penalty. Literature survey indicates that Deep Neural Networks (DNN) have been extensively used for diagnosis and detection of COVID-19 from chest CT images. After thoroughly studying the published literature, we observe a few research gaps. Though several studies have achieved substantial classification accuracies, their performance depends heavily on a large annotated dataset and suffers from mis-prediction on

smaller datasets. A few other research works could predict well on a small hospital dataset but could not perform as well on external data. Researchers have evaluated existing Deep CNN models to identify COVID-19 from chest CT images. However, automatic feature extraction by DNNs lack interpretability, which is critical in health care diagnostics. Requirement of a very large annotated dataset, high computational resources in terms of sophisticated Graphics Processing Units (GPUs), memory, training time etc. are primary concerns in using DNNs. Published literature on image texture analysis show that abnormal lung anatomy (caused due to various lung ailments) exhibit visual patterns that are distinct from those of normal and healthy lungs. Published studies suggest that shape and texture features characterize the most prevalent patterns related to lung abnormalities. Empirical studies indicate better performance of second and higher order statistical approaches as compared to the first order statistical features [28]. As texture parameters indicate variations in pixel intensity values of chest CT images, we propose a hybrid texture feature extraction approach followed by an efficient feature reduction technique to select the most relevant features to distinguish between COVID-19 and non-COVID-19 images. A lightweight deep CNN optimized by a modified bio-inspired optimizer is used in the final classification stage.

3. Methodology

The proposed model consists of four major phases viz., image acquisition, image preprocessing, feature extraction, feature selection and classification. The input images obtained from an open-source database of chest CT scans, COVID-CT are resized to a uniform size and filtered to eliminate noise. Region-based segmentation is employed to segment lungs from the chest CT images. Haralick features, Local Binary Pattern (LBP) codes, and Local Directional Pattern (LDP) codes are computed to obtain texture and intensity features. A pre-trained ResNet-101 network is employed in the next stage to extract more complex and subtle features. ReliefF feature selection algorithm selects the most relevant features thereby eliminating redundant features and reducing dimensionality. The most relevant features are classified into COVID-19 and non-COVID-19 by a 7-layered Deep Convolutional Network. The Deep Convolutional Network is optimized by a novel Rapid Navigation Optimizer, that is a hybrid of Moth-Flame Optimizer (MFO) and Mayfly Algorithm (MA). The stages of the proposed model are discussed in detail in the sub-sections 3-A to 3-F. The proposed methodology is illustrated in Fig. 2.

A. Image Acquisition:Dataset Details

Chest CT scans for training and testing the proposed model were collected from a public dataset, COVID-CT, created by Yang, Zhao, et al. [10]. This dataset contains 349 CT images of COVID-19 belonging to 216 patients and 397 CT images that are negative for COVID-19 and are treated as non-COVID-19. Yang, Zhao, et al. [10] compiled the set of non-COVID-19 images from four open – access datasets, the MedPix database, the LUNA dataset, the Radiopaedia website and the PubMed Central (PMC) archive. All images are in PNG format and of variable sizes. The dataset is available on https://github.com/UCSD-AI4H/COVID-CT.

A second larger public dataset, the COVIDx-CT-3A was utilized to test the effectiveness of the proposed Rapid Navigation Optimizer algorithm. COVIDx-CT-3A is an open access benchmark dataset shared by Hayden Gunraj [29]. The dataset is created by compiling several open datasets. It comprises 425,024 CT slices from 5,312 patients. It contains 310,593 COVID-19 CT images and 71,488 CT images of normal people. The open access dataset was last updated in June 2022 and is available on https://kaggle.com/datasets/hgunraj/covidxct.

B. Image Pre-processing

The steps involved in image preprocessing are as follows:

- Image resizing: As images obtained from the public dataset are of variable sizes, all images are resized to 256x256 pixels.
- Image enhancement: Unwanted noise in images is eliminated by use of Gaussian filters.
- ROI extraction: Region-based segmentation that exploits similarity of neighborhood pixels is used to extract the lung region from the chest CT image.

C. Feature Extraction

As texture parameters indicate variations in pixel intensity values of chest CT images, texture features of chest CT scans can be effectively exploited to detect COVID-19 lesions. The proposed model employs the following hybrid feature extraction techniques, (i) GLCM-based Haralick texture features (ii) Local Binary Pattern (LBP) - based features (iii) Local Directional Pattern (LDP) – based texture features (iv) Directional Local Binary Pattern ((DLBP) (v) Fusion of LDP and LBP. The features obtained are converted into virtual RGB format images. The virtual RGB format images of computed features are input to a pre-trained ResNet101 neural network for further extraction of features. The bi-stage feature extraction process is as follows:

1) Stage 1: Texture feature computation

1) Step 1: Computation of GLCM-based Haralick texture features.
   Grey Level Co-occurrence Matrix (GLCM), a second-order statistical approach specifies the relative frequencies $P(i, j)$ with which two neighboring pixels with grey level intensity (i, j) occur at a constant vector distance (d, θ) from each other in the image. Maximum texture information is contained in a GLCM and hence it is the basis for computing the fourteen Haralick features that relate to textural

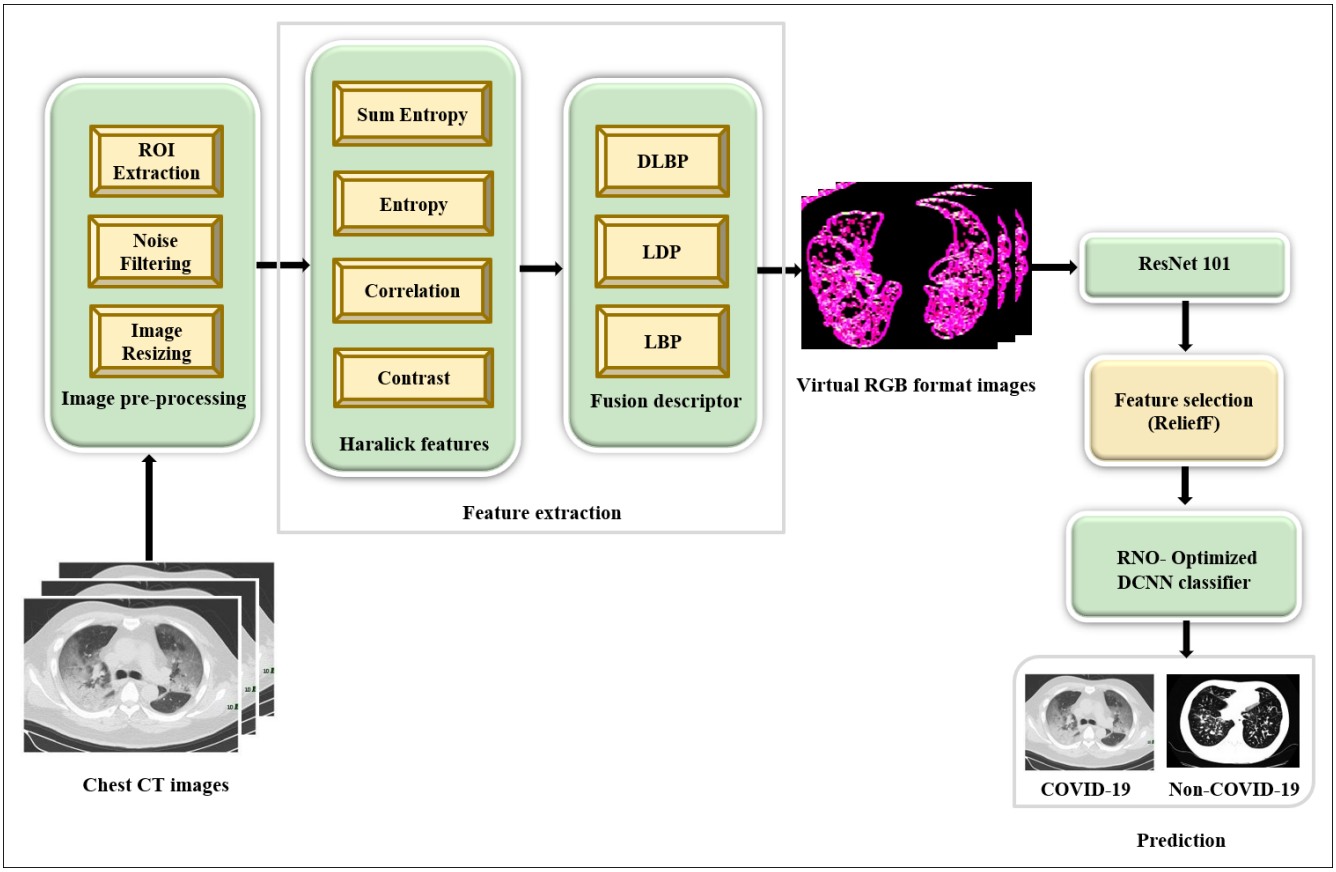

Figure 2. Proposed methodology

characteristics of an image [30]. As the Haralick features are highly correlated, we compute ten Haralick features. ANOVA and Chi-squared feature analysis tests on the computed features indicate that contrast, entropy, sum entropy and correlation are the highest ranked features. Mathematical equations for these four Haralick features are listed in Table I. Contrast, entropy, sum entropy and correlation values are mapped to 3 color planes to create a virtual RGB format image.

2) Step 2: Computation of a joint feature descriptor comprising of (i) Local Binary Pattern (LBP), (ii) Local Directional Pattern (LDP) codes and (iii) Directional LBP (DLBP)

TABLE I. Haralick Features

Feature	Formula
Contrast	$f_2 = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lvert i-j \rvert^2 \times p(i,j)$
Correlation	$f_3 = \frac{\sum_i \sum_j (ij)p(i,j) - \mu_x \mu_y}{\sigma_x \sigma_y}$
Sum Entropy	$f_8 = -\sum_{i=0}^{2N} p_{x+y} log(p(i))$
Entropy	$f_9 = -\sum_i \sum_j p(i,j) log(p(i,j))$

a) Local Binary Pattern (LBP):
The Local Binary Pattern (LBP) operator is a function of the change of intensity around the pixel to encode the microlevel information of spots, edges, and other local features in the image. The intensity of the center pixel of a 3x3 window in the image is used as a threshold to compare against the neighboring pixels [31]. The LBP code is computed in steps (i), (ii) and (iii).

i) The LBP code, $LBP_N(x,y)$ is computed using equation 1 on an image of size RxC for each pixel $p(x,y)$, where x and y are in the range, $0 \le x \le R$ and $0 \le y \le C$. g_i and g_p are the gray levels of pixel p and its i^{th} neighbor respectively.

$$LBP_N(x,y) = \sum_{i=0}^{N-1} S(g_i - g_p) \times 2^i \quad (1)$$

ii) The binary function $S(x,y)$ is computed

using equation 2.

$$S(x,y) = \{1\}, if\, x \geq 0; S(x,y) = \{0\}, otherwise \quad (2)$$

iii) Binary codes obtained in step (ii) are converted into decimal.

b) Local Directional Pattern (LDP)
Local Directional features are considered by computing the edge response values in different directions. As the LDP descriptor considers the edge response values in different directions instead of only pixel intensities, it is more robust against different variations like non-monotonic changes in illumination and random noise [31]. The LDP code is computed in steps (i), (ii), (iii), (iv), (v) and (vi).

i) Eight directional responses of pixels are computed using the Kirsch compass edge detector in eight orientations ($M_7 \ldots M_0$) centered on its own position. The 8 directional Kirsch masks, East, North-East, North, North-West, West, South-West, South and South-East, are represented by equations 3 to 10, respectively. For each direction i, the i^{th} directional response m_i of every pixel (x,y) of the image is computed using equation 11 by applying the corresponding mask M_i.

$$\begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \quad (3)$$

$$\begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \quad (4)$$

$$\begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad (5)$$

$$\begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \quad (6)$$

$$\begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \quad (7)$$

$$\begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \quad (8)$$

$$\begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \quad (9)$$

$$\begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} \quad (10)$$

$$m_i = \sum_{k=-l}^{1} \sum_{l=-1}^{1} M_i(k+1, l+1) \times I(x+k, y+l) \quad (11)$$

ii) The eight directional responses ($m_0 ... m_7$) obtained in step (i) are sorted in order of their values.

iii) Generation of the LDP code is based on the k most significant directional responses. The corresponding bits representing most significant directional responses are set to binary 1 and remaining 8-k bits to 0 for $k = 3$. The value, $k = 3$ has been widely used in published literature

iv) The LDP code, $LDP_{x,y}(m_0, ..., m_7)$, of the pixel (x,y) with directional responses $(m_0 \ldots\ldots m_7)$ and m_k, the k^{th} most significant response is computed using equation 12.

$$LDP_{x,y}(m_0, ..., m_7) = \sum_{i=0}^{7} s(m_i, ...m_k) \times 2^i \quad (12)$$

v) The binary function $S(x)$ is computed by equation 13.

$$S(x,y) = \{1\}, if\, x \geq 0; S(x,y) = \{0\}, otherwise \quad (13)$$

vi) Binary LDP codes obtained in step (v) are converted into decimal.

c) Directional Local Binary Pattern (DLBP)
The third component of the joint feature descriptor, the Directional Local Binary Pattern (DLBP) code is a combination of LDP and LBP. DLBP eliminates the dependency of LDP code on the selection of value of number of significant bits, k. It is computed in three steps (i), (ii) and (iii) as follows:

i) Gradient directional responses, $m_0, ..., m_7$ are computed by applying the Kirsch masks represented by equations 3 to 10 and using equation 11.

ii) Binary DLBP code for each pixel (x,y) is generated by comparing the value of the center pixel, $v(x,y)$, to the response values, $m_0, ..., m_7$ obtained in step (i). The DLBP code is computed as shown in equation 14.

$$DLBP_{x,y}(m_0, ..., m_7) = \sum_{i=0}^{7} S(m_i - v(x,y)) \times 2^i \quad (14)$$

iii) The binary DLBP codes obtained in step (ii) are converted into decimal.
iv) A joint feature descriptor is generated by a fusion of decimal codes of LBP, LDP and DLBP and converted into virtual RGB image format.

2) Stage 2: ResNet101-based Feature Extraction

As the public dataset used is small, a pre-trained ResNet101 is employed. ResNet101, a variant of the baseline residual network has total 347 layers, corresponding to 101 fully connected layers on the path from the input layer to the output layer. It has been trained on a subset of the ImageNet database, and can classify images into 1000 object categories [32]. To extract more complex and subtle features, virtual RGB format images of the computed Haralick features (Contrast, Entropy, Sum Entropy and Correlation) and the joint feature descriptor (LBP, LDP and DLBP) are input to a pre-trained ResNet 101 neural network. The images are resized to 224x224x3 to fit requirements of ResNet 101 and 'activations' with Mini Batch size of 32 is used to extract feature representations from 'fc1000' layer of ResNet101. Total number of features extracted by the ResNet101 in the proposed model are of the dimensions 544x5001.

D. ReliefF – based Feature Selection

As the feature set generated by ResNet101 is of high dimension (544x5001features), a feature selection algorithm is applied to eliminate redundant and irrelevant features. The feature selection algorithm also serves to reduce dimensionality and improve classification performance. The Relief feature selection algorithm calculates a proxy statistic, referred to as feature weights W_f for each feature f. The feature weight is used to estimate the feature relevance. The ReliefF feature selection algorithm is the sixth and the best-known variant of the baseline Relief algorithm. It is computationally efficient and sensitive to complex patterns of associations. ReliefF relies on a user parameter k indicating number of neighbors. k specifies the use of k nearest hits and k nearest misses in the scoring update for each target instance (rather than a single hit and miss) [33].

Considering k number of neighbors has increased weight estimate reliability, particularly in noisy problems. Based on preliminary empirical testing a value of 10 has been widely adopted as the default setting for k number of neighbors. In the proposed study, the feature vector of dimensions 544 x5001 generated by ResNet 101 are input to the ReliefF feature selection algorithm. Value of k, the number of neighbors is 10. ReliefF returns 1000 highest ranked features. Steps involved in the ReliefF algorithm are as follows:

1) *Step 1: Initialization of feature weights*
Initialize all feature weights, W_f to zero.
2) *Step 2: Determination of hits and misses of an instance x*
Randomly select an instance x and find k nearest neighbors that belong to the same class, referred to as nearest hits $H_j(x)$ and different classes, referred to as nearest misses $M_j(x)$.
3) *Step 3: Updation of the quality estimator W_f*
Update the value of the quality estimator, W_f for all attributes f by considering their values for x, $H_j(x)$ and $M_j(x)$ as follows:
   a) If the values for x and $H_j(x)$ are different for attribute f (the attribute f separates the two instances of same class) then decrease the value of W_f.
   b) If the values of x and $M_j(x)$ are different for attribute f (the attribute f separates the two instances of different class) then increase the value of W_f. Weight updating formula is specified in equation 15.

$$W_f^{i+1} = W_f^i + \sum_{c \neq class(x)} \frac{\frac{p(x)}{1-p(class(x)}\, diff(x, m_j(x))}{m} \tag{15}$$

   where diff() is the distance between two samples of the feature f when finding nearest neighbors and $p(x)$ is the probability of a class.
   c) Euclidian distance is used to compute the inter-class and intra-class distances of samples and is specified in equation 16.

$$D(x, y) = \sqrt{\sum_{i-1}^{N} (x_i - y_i)^2} \tag{16}$$

4) *Step 4: Determining the relevance of features*
Sort the feature indices in ascending order of their weights (relevance).
5) *Step 5:Selection of most relevant features*
Select the 1000 highest ranked features.

E. Optimized Deep Convolutional Neural Network (ODCNN)

The final stage of the proposed model involves classification of COVID-19 and non-COVID images based on the selected features. The highest ranked 1000 features selected through the ReliefF feature selection algorithm are input to an Optimized Deep Convolutional Neural Network (ODCNN). The Optimized Deep Convolutional Neural Network (ODCNN) is trained for 30 epochs, and a learning rate of 0.001. Architecture of the proposed DCNN is illustrated in Fig. 4. The ODCNN comprises of 7 layers, as follows:

- Layer 1: 2-D input layer of size 1000 x 1.
- Layer 2: The 2-D Convolution layer comprising of 5 filter kernels of size 1x1.
- Layer 3: The Rectified Linear Unit (ReLU) is an activation layer and applies non-linearity to the convolution output.

- Layer 4: 2-D max-pooling layer extracts the maximum value in a 1x1neighborhood with a stride equal to 5. The flattening layer following the 2-D max-pooling operation flattens its output into a single array of dimension 1000.

- Layer 5: Fully Connected Layer (FC) comprises 2 output neurons. Weights extracted from layer 5 are fine - tuned using various optimizers such as, Adam optimizer (inbuilt), Moth-flame optimizer (MFO), Mayfly optimizer and the proposed Rapid Navigation Optimizer (RNO). Performance analysis of these optimizers is presented in Section 4-C.

- Layer 6: Softmax Layer outputs a vector with probabilities of each possible outcome. (COVID-19/ Non-COVID-19). It is mathematically represented by equation 17.

$$S(y_i) = \frac{exp(y_i)}{\sum_{j=1}^{n}(y_i)} \tag{17}$$

  where y is the input vector to the softmax function, S. It consists of n elements for n classes. y_i is the i^{th} element of the input vector, y. $\sum_{j=1}^{n}(y_i)$ is a normalization term.

- Layer 7: Final classification layer infers the output class, class 1 for COVID-19 and class 0 for non-COVID-19.
  Details of input features, layer wise activation, filters and learnable parameters are listed in Table II. The 7-layered ODCNN has total 2K learnable parameters.

The ODCNN performance is evaluated for the following optimizers: (1) Adam optimizer (inbuilt) (2) Moth-flame optimizer (MFO) (3) Mayfly optimizer (4) The proposed Rapid Navigation Optimizer (RNO).

F. Proposed Rapid Navigation Optimization Algorithm

Fine-tuning weights and bias of the proposed ODCNN to maximize classification accuracy and to prevent overfitting is an optimization problem. Nature-inspired optimization algorithms have been successfully used to solve many engineering problems. A nature-inspired optimization approach, the Rapid Navigation Optimization (RNO) algorithm is proposed to tune the weights of the DCNN. It integrates the attributes of the well-known Mayfly optimization technique and the Moth-Flame Optimizer (MFO) algorithm. The nature – inspired optimization algorithms considered for the proposed Rapid Navigation Optimization (RNO) algorithm are discussed in brief in Sec. F.1 and Sec. F.2. The mathematical model of the proposed Rapid Navigation Optimization (RNO) algorithm is explained in Sec. F.3. Algorithm of the proposed Rapid Navigation Optimization (RNO) is shown in Sec. F.4.

1) *Overview of Mayfly Algorithm (MA)*
The Mayfly algorithm is an improved modification of the Particle Swarm Optimization (PSO) algorithm. It offers a powerful hybrid algorithm structure by combining the major benefits of Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). The mayflies exhibit a typical social behavior, particularly during their mating/breeding process. Mayfly algorithm draws inspiration from their mating process. The male mayflies congregate in large numbers a little above water. They perform a nuptial dance to attract the female mayflies. As male mayflies always gather in swarms, the position of each male mayfly is influenced by its own personal experiences and the experiences of the neighboring male flies [34]. A female mayfly individually flies to a male fly to breed. The crossover operator represents the mating process. Design of crossover operator ensures that the best male breeds with best female mayfly. Initially two sets of mayflies, the male and female population are randomly generated. The potential solution is obtained by randomly placing each mayfly in the problem space. Each mayfly is randomly placed in the problem space as a potential solution. This is represented by a d – dimensional vector, $x = (x_1, ..., x_d)$. Its performance evaluation is done by a predefined objective function, f(x). Velocity of a Mayfly is defined as change of its position, $v = (v_1, v_2, ..., v_d)$. The dynamic interaction of both *pbest* and *gbest* at time t determines the direction in which a mayfly flies. The global best (*gbest*) solution is the best position attained by any mayfly in the swarm so far and depends on *pbest*. Change in position of a male mayfly i at time step t is formulated by adding a velocity component v_i^{t+1} to the current position, x_i^t, as shown in equation 18.

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{18}$$

Male mayflies are assumed to move constantly. They cannot develop higher speeds as they are always a few meters above water performing the nuptial dance. Accordingly, the velocity of male mayfly, v_{ij}^{t+1} is computed as specified in equation 19.

$$v_{ij}^{t+1} = v_{ij}^t + a_1 e^{-\beta r_p^2}(pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2}(gbest_{ij} - x_{ij}^t) \tag{19}$$

v_{ij}^t - velocity of mayfly i in the dimension j = 1,...,n at time step t; x_{ij}^t - position of mayfly i in the dimension j = 1,...,n at time step t; a_1 and a_2 – Positive attraction co-efficient to scale the contribution of the cognitive and social component; β- Fixed visibility co-efficient; r_p – Cartesian distance between x_i and $(pbest)_i$; r_g – Cartesian distance between x_i and $gbest$. Each mayfly adjusts its path of flight towards its personal best position (*pbest*) so far and the best position attained by any mayfly of the swarm so far (*gbest*), thus ensuring optimal solution.

2) *Overview of the Moth-Flame Optimizer (MFO)*
The transverse orientation navigation mechanism of moths is the main source of motivation to develop

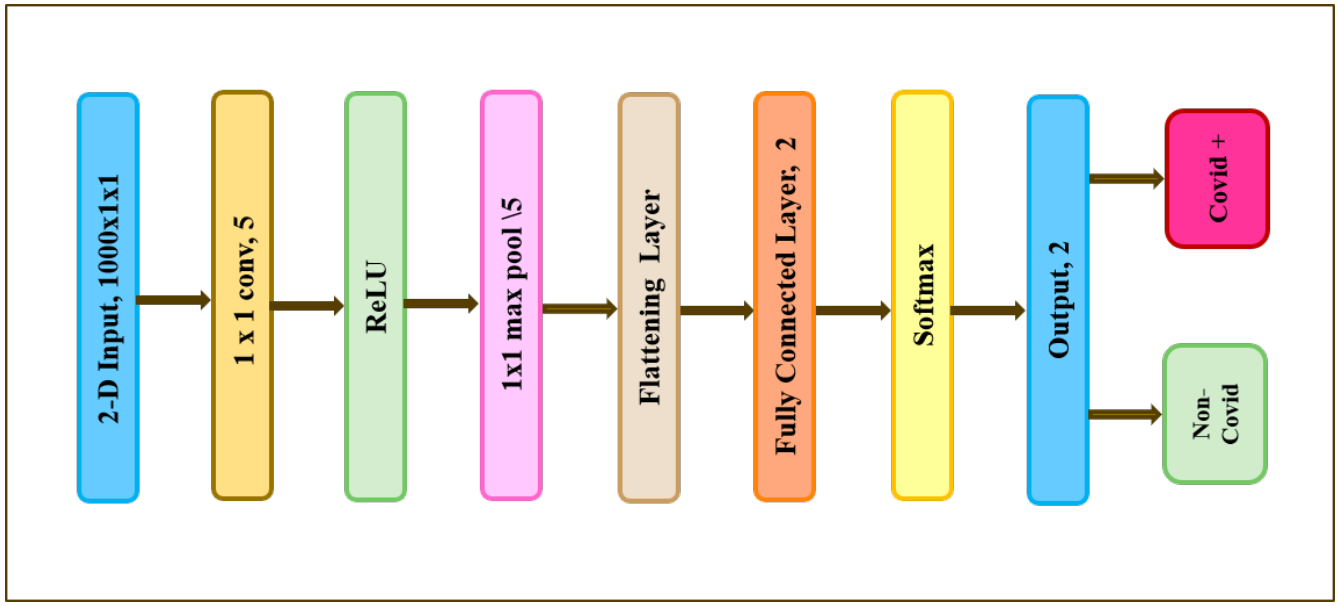

Figure 3. Optimized Deep Convolutional Neural Network

TABLE II. ODCNN layer and activation details

Layer No.	Layer Name	Type	Activations	Learnable Parameters
1	**imageinput** 1000x1 images with 'zerocenter' normalization	Image Input	1000(S)x1(S)x1(C)x1(B)	-
2	**conv** 5 1x1x1 convolutions with stride [1 1] and padding [0 0 0 0]	2-D Convolution	1000(S)x1(S)x5(C)x1(B)	Weights 1 x 1 x1 Bias 1 x 1 x 5
3	**relu** ReLU	ReLU	1000(S)x1(S)x5(C)x1(B)	-
4	**maxpool** 1x1 convolutions with stride [5 5] and padding [0 0 0 0]	2-D Max Pooling	200(S)x1(S)x5(C)x1(B)	-
5	**fc** 2 fully connected	Fully Connected	1(S)x1(S)x2(C)x1(B)	Weights 2x1000 Bias 2x1
6	**softmax** softmax	Softmax	1(S)x1(S)x2(C)x1(B)	-
7	**classoutput** crossentropyex with classes '1' and '2'	Classification Output	1(S)x1(S)x2(C)x1(B)	-

the Moth-Flame Optimizer (MFO) algorithm. Hence, the MFO algorithm is classified as a nature-inspired algorithm. The moths use this mechanism to fly in the night using the moon light. Moths fly by maintaining a fixed angle with the moon. Fig. 5 illustrates the concept of transverse orientation. However, the transverse orientation is effective only when moving in a straight line when the light source is very far. When a moth sees a light at a close distance, it tries to fly at a similar angle with the light and ends up flying in a vicious helical path. It eventually converges towards the artificial source of light. Fig. 5 depicts the spiral flying path of moths around close light sources. Seyedali Mirjalili developed a mathematical model of the spiral flying path of moths around artificial lights and proposed the Moth-Flame Optimizer (MFO) [35]. Moths are the candidate solutions and the problem's variables

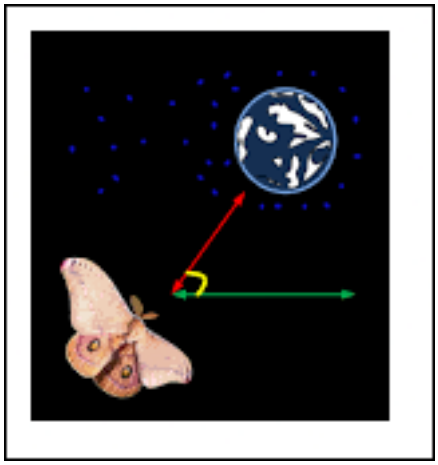

Figure 4. Transverse orientation navigation

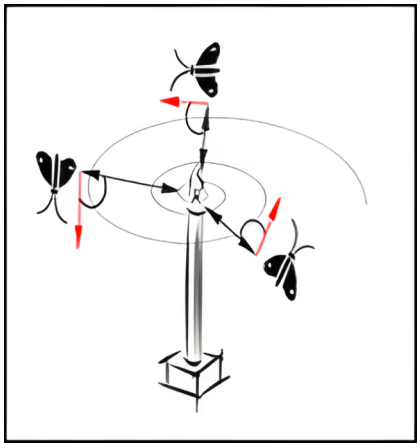

Figure 5. Spiral motion of moths

are the position of moths in space. The key concepts of MFO algorithm are as follows:

a) MFO is a population-based algorithm. The matrix M, specified in equation 20 represents the initial position of each moth in the moths' population.

$$M = \begin{pmatrix} m_{1,1} & \dots & m_{1,d} \\ \dots & \dots & m_{2,d} \\ m_{n,1} & \dots & m_{n,d} \end{pmatrix} \quad (20)$$

where n is the population size (number of moths); d represents the number of variables/dimensions.

b) Fitness value of each moth is computed after obtaining positions of all moths in the search space. The fitness value of each moth is calculated using a fitness (objective) function, f(obj) and the result is saved in the fitness matrix OM, represented by equation 21.

$$OM = [om_1 om_2 ... om_n]^T \quad (21)$$

c) The variable, 'flame' represents the moth's best position (best solution) that has been found so far [35]. The 'flame' positions and their fitness values are stored in the matrices F and OF as indicated by equations 22 and 23 respectively.

$$F = \begin{pmatrix} f_{1,1} & \dots & f_{1,d} \\ \dots & \dots & f_{2,d} \\ f_{n,1} & \dots & f_{n,d} \end{pmatrix} \quad (22)$$

$$OF = [of_1 of_2 ... of_n]^T \quad (23)$$

Moths explore the search space, while flames are the best positions of moths, obtained so far. The matrix F in equation 22 always represents the n recent best solutions obtained so far.

d) The transverse orientation is mathematically modelled by a logarithmic spiral. A moth represents the initial and a flame indicates the final point of the spiral. The logarithmic spiral, space around the flame and positions at different values of t is illustrated in Fig. 7. The spiral flying path of moths is mathematically defined in equations 24, 25 and 26.

$$M_i = S(M_i, F_i) \quad (24)$$

M_i is the updated position of the i^{th} moth; S describes a logarithmic spiral curve; F_j represents the j^{th} flame

$$S(M_i, F_j) = D_i \times e^{bt} \times cos(2\pi t) + F_j \quad (25)$$

b is a constant that defines the shape of the logarithmic spiral;

$$D_i = |F_j - M_i| \quad (26)$$

D_i is the distance between the i^{th} moth and the j^{th} flame. Equation 25 clearly indicates that the next position of a moth is defined with respect to a flame. The spiral motion ensures that a moth flies around the flame and guarantees exploration and exploitation of the search space. To further emphasize exploitation, it is assumed that t is a random number between [a,1], where a decreases from -1 to -2 as the number of iteration increases. At each iteration, and after updating the list of flames, the flames are sorted based on their fitness values. The moths then update their positions with respect to the best flames. Accordingly, the first moth always updates its position with respect to the best flame and the last moth updates its position with respect to the worst flames. As the moths move around different flames there is higher exploration of the search space and lower probability of local optima stagnation.

e) To improvise exploration of the optimal solu-

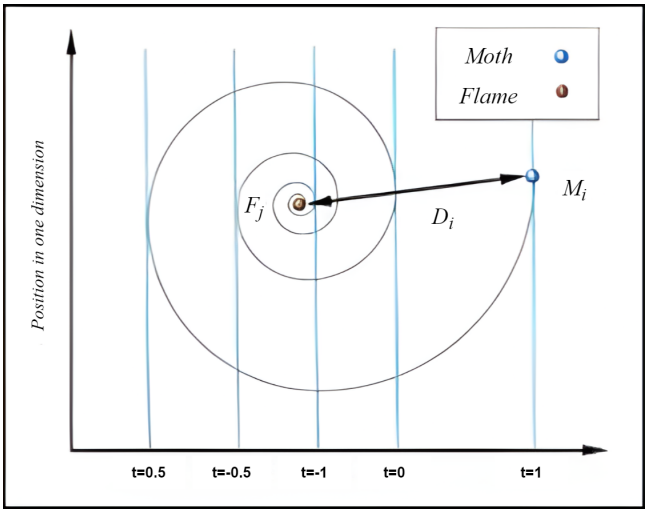

Figure 6. Spiral path of Moths

tion and enhance the convergence rate of the algorithm in the later stages, there is a need to adaptively reduce the number of flames [36]. The number of flames is reduced adaptively over the course of iterations and is specified by equation 27.

$$N_{flame} = round(N - l \times \frac{N-1}{T}) \quad (27)$$

N represents the maximum number of flames; l is the current number of iterations and T indicates the maximum number of iterations. The search performance of the MFO relies on exploration (algorithm searches the whole space) and exploitation (local search in a small search area).

3) *Mathematical Model of the Proposed Rapid Navigation Optimizer (RNO) Algorithm*

The original Mayfly Algorithm (MA) [34] and MFO algorithm [35] are used as the baseline for deriving the equations of the proposed Rapid Navigation Optimizer (RNO) Algorithm. To avoid local optimum stagnation and maintain balance between exploration and exploitation, concept of random radius search is introduced. The concept of velocity associated with the Mayfly Algorithm (MA) is integrated into the position renovation stage of the Moth fly to improvise the effective region search and maximize the speed of convergence. The four significant stages involved in the proposed RNO algorithm are as follows:

a) *Stage 1: Initialization*

Population position of moths represent the initial weights. Population position of moths in the search space are initialized as specified in equation 28.

$$M = R(n,d) \times (ub - lb) + lb \quad (28)$$

where ub and lb are the upper and lower bounds of the search space, respectively; n is the population size, representing the number of moths; d represents the number of variables/dimensions. In the proposed algorithm, the number of variables are equal to weights in layer 5 of the fully-connected layer of the Optimized Deep Convolutional Neural Network (ODCNN); R is the random number generation function, that generates random numbers with a uniform distribution in the range (0,1) [35].

b) *Stage 2: Exploration stage*

i) Initial position corresponding to each moth is stored in matrix M, specified in equation 20.

ii) The fitness value of each moth is calculated using a fitness (objective) function, f(obj). Result of fitness function is saved in the fitness matrix OM, represented by equation 21. The fitness function, f(obj) in the proposed algorithm specifically calculates the accuracy of the ODCNN classifier after updating the weights in layer 5. Accuracy is computed using equation 33. The fitness value is the accuracy returned by the objective function, f(obj) and must be maximized to improve performance of the ODCNN.

iii) The 'flame' positions and their fitness values are stored in the matrices F and OF as indicated by equations 22 and 23 respectively.

c) *Stage 3: Position renovation stage*

This stage models the spiral motion of the moths in real life [35]. The spiral flying path of moths is mathematically described in equations 24, 25 and 26. The number of flames is decreased adaptively over the course of iterations and is specified by equation 27.

d) *Stage 4: Effective region determination*

i) In the proposed Rapid Navigation Optimizer (RNO), velocity and search space are considered significant for updating the effective position of the moths. Equation 29 describes selection of search space and updating of the foraging velocity around the flame of the moth fly.

$$X_{t+1}^{vr} = S(M_i, F_j) + v_{ij}^{t+1} + e(\tau, X^t) \quad (29)$$

e – Moth fly exploring the search space; τ - Search Radius; $S(M_i, F_j)$ - spiral motion of the Moth fly; v_{ij}^{t+1} – velocity of male mayfly.

ii) To improvise the effective region search during position renovation of the Moth fly, concept of velocity associated with the male mayfly is applied to the standard MFO algorithm. The spiral motion of the moth fly ensures that a moth flies around the flame and guarantees exploration and exploitation of the search space. In the Mayfly Algorithm, velocity of male mayfly, v_{ij}^{t+1} is computed as specified in equation 19. Velocity of the male mayfly and hence its position is a function of the personal best position (*pbest*) a mayfly has ever visited and the best position attained by any mayfly of the swarm so far (*gbest*). As this concept ensures an optimal solution in the Mayfly Algorithm, it is incorporated in the spiral motion of the MFO to enhance optimization results. Accordingly, equation 19 of the mayfly is substituted in equation 29 as shown in 30.

$$X_{t+1}^{ma} = e(\tau, X^t) + v_{ij}^t + a_1 e^{-\beta r_p^2}(pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2}(gbest_{ij} - x_{ij}^t) \quad (30)$$

X_{t+1}^{ma} represents the updated solution of the Mayflies.

iii) The final position update rule for the proposed Rapid Navigation Optimization (RNO) algorithm follows the characteristics of the mothfly and mayfly. Hybridizing characteristics of both flies with equal significance, we obtain a better solution in terms of exploration, exploitation and the best solutions obtained during iterations. The final position update rule for the proposed Rapid Navigation Optimization (RNO) is modelled in equations 31 and 32.

$$X^{t+1} = \frac{1}{2}[S(M_i, F_j) + X_{t+1}^{ma}] \quad (31)$$

$$X^{t+1} = \tfrac{1}{2}[D_i \times e^{bt} \times cos(2\pi t) + F_j + e(\tau, X^t) + X^t + e(\tau, X^t) + v_{ij}^t + a_1 e^{-\beta r_p^2}(pbest_{ij} - x_{ij}^t) + a_2 e^{-\beta r_g^2}(gbest_{ij} - x_{ij}^t)] \quad (32)$$

Equation 32 provides a better solution during the rapid search in the exploration space. The classification accuracy of the RNO-based ODCNN for the K-fold value of 10 is 97.260%. There is a performance improvement of 4.19% over the accuracy of the standard MFO – based ODCNN. A performance improvement of 6.71% is indicated in the classification accuracy of the RNO-based ODCNN optimized over the accuracy of the ODCNN when optimized using the vanilla Mayfly Algorithm (MA).

4) *Algorithm of Proposed Rapid Navigation Optimizer (RNO)*
The algorithm for the proposed Rapid Navigation Optimizer is depicted in Table III.

4. Performance Evaluation of the Proposed Optimized Convolutional Neural Network

The performance of our proposed ODCNN model and other comparative methods are described in this section. For the performance evaluation and comparative analysis, the metrics computed were accuracy, precision, sensitivity/recall, specificity, F1-Score and Receiver Operating Characteristics (ROC). The performance metrics were computed with respect to training percentage and K-fold validation. The proposed ODCNN model was evaluated on 3 types of datasets:

1) The COVID-CT dataset [10]
2) Augmented COVID-CT dataset
3) COVIDx-CT-3A dataset [29]

A. Experimental Setup and Parameter Settings

All training and test simulations were implemented using MATLAB software on a 1.6 GHz Intel Core i5-8265U Quad-Core laptop with 8GB DDR4, 16GB Optane, 1TB HDD and Windows 10 operating system. The size of the input layer of the ODCNN was set to 1000, that is the size of the features selected by ReliefF algorithm. The output layer comprises 2 nodes representing two classes, COVID-19 and non-COVID-19.

The ODCNN performance was evaluated for the following optimizers: (1) Adam optimizer (inbuilt) (2) Moth-flame optimizer (MFO) (3) Mayfly optimizer (4) The proposed Rapid Navigation Optimizer (RNO). Values of number of search agents, the moths (n) and maximum number of iterations (T_{max}) for the proposed RNO algorithm were set empirically. The optimal values of both n and T_{max} that achieved maximum classification accuracy was 5.

B. Performance Metrics

The performance measures considered for analyzing the proposed RNO algorithm and the other existing methods are described below.

1) Accuracy (Acc)

The number of accurate predictions of positive and negative cases from the disease patient's test samples to the ratio of the total number of provided test samples. Classification accuracy is computed as specified in 33.

$$\frac{Accurate\ predictions\ of\ positive\ and\ negative\ cases(TP + TN)}{Total\ test\ samples(TP + TN + FP + FN)} \quad (33)$$

TABLE III. Rapid Navigation Optimizer Algorithm

Rapid Navigation Optimizer

INPUT: Weights from layer 5 of the 7-layered ODCNN
OUTPUT: Best_Solution (Optimal weights)
PARAMETERS:
n: Number of Moths
T: Current iteration; T_{max} : Maximum number of iterations
d: Number of variables (dimensions) = Number of weights in Layer 5
N: Maximum number of flames
BEGIN
Set the initial values of lower bound (lb), upper bound (ub) of the search space as per values of weights obtained from layer 5
Define objective function fobj: Compute Accuracy of ODCNN after updating weight using equation 33
Randomly initialize moth positions (M) as specified in equation 28
While T ¡ T_{max}:
  Compute number of flames (N_{flame}) using equation 27
  Check and correct the moth positions with respect to lower bound (lb), upper bound (ub) of the search space
  Evaluate fitness of each moth using $f_{obj} : OM = f_{obj}(M)$
  If T==1:
    [Fitness_sorted , I] = sort (Fitness of 1st population of moths)
    sorted_population = Moth positions, M as per I
    Flames (F) = sorted_population
    Compute flame (F) fitness and sort flames as per their fitness values: OF = sort(F)
  else:
    F = sort(M_{T-1}, M_T); T is current iteration, T-1 is the previous iteration
    OF = sort(OM_{T-1}, OM_T)
  end
end
for i=1:n
  for j=1:d
    Update 't' in equation 25 between [a,1] where a decreases from -1 to -2 as the number of iterations increases
    Compute Di using equation 26 with respect to the corresponding moth
    Update moth position in accordance to equation 32
  end
end
Best_flame_pos = Best_Solution (weights)
END

2) Recall / sensitivity (Sen)

The accurate prediction of positive cases from the diseased patient's test samples to the ratio of accurate prediction of positive cases and inaccurate prediction of negative cases from the test samples. Sensitivity, also termed Recall is calculated using equation 34.

$$\frac{Accurate\ prediction\ of\ positive\ cases(TP)}{Accurate\ positive\ and\ inaccurate\ negative\ prediction(TP+FN)} \quad (34)$$

3) Specificity (Spe)

The accurate prediction of negative cases from the diseased patient's test samples to the ratio of accurate prediction of negative cases and inaccurate prediction of positive cases from the test samples. Specificity is computed as specified in equation 35.

$$\frac{Accurate\ prediction\ of\ negative\ cases(TN)}{Accurate\ negative\ and\ inaccurate\ positive\ prediction(TN+FP)} \quad (35)$$

4) Precision (Pre)

The accurate prediction of positive cases from the diseased patient's test samples to the ratio of total number of positive predictions from the test samples. Precision is computed as specified in equation 36.

$$\frac{Accurate\ prediction\ of\ positive\ cases(TP)}{Total\ number\ of\ positive\ prediction(TP+FP)} \quad (36)$$

5) F1-score (F1)

F1-score is the harmonic mean of Precision and Recall/ Sensitivity. Equation 37 specifies F1-score computation.

$$F1 = 2 \times [\frac{Pre \times Sen}{Pre + Sen}] \quad (37)$$

The parameters in the computations are as follows:

1) TP: True Positive
2) TN: True Negative
3) FP: False Positive
4) FN: False Negative

C. Comparative Performance Analysis of the Various Classifiers Implemented for COVID-CT Dataset

The comparative performance analysis of the classifiers implemented is based on the following two criteria; (A) K-fold cross - validation for K-values of 4, 6, 8 and 10. (B) Training data percentage of 40, 60 and 80.The performance analysis presented in this section is based on COVID-CT dataset. The 1000 highest ranked features selected by ReliefF algorithm were input to the following classifiers:

1) k-Nearest Neighbors (KNN):
   A supervised machine learning classifier, the k-nearest neighbors (KNN) algorithm relies on the distance between the query point and the other data-points in the training dataset to make classifications. The algorithm identifies the 'k' nearest neighbor of the query point. The most common class label among the 'k' neighbors is assigned to the query point. To evaluate the performance of KNN classifier in this study, we utilized Euclidean distance metric. We set k = 4, where k value in the k-NN algorithm defines the number of neighbors checked to determine the classification of a specific query point.
2) Support Vector Machine (SVM):
   We evaluated performance of SVM by inputting the 1000 highest ranked features and corresponding label data vector as the training data.
3) Feed Forward Neural Network (NN):
   Feed forward Neural Networks propagate information in the forward direction in the network through the input nodes. We input the 1000 highest ranked features and corresponding label data vector as the training data. We evaluated the neural network with following specifications, one hidden layer of size 10, Levenberg-Marquardt training function and 1000 epochs.
4) Pre-trained AlexNet Neural Network:
   AlexNet is a 8 layers deep convolutional neural network. We evaluated a pretrained version of the network in our study.
5) 7-layered ODCNN with in-built Adam optimizer
6) Mayfly optimized ODCNN
7) Moth flame optimized ODCNN
8) Proposed Rapid Navigation Optimization – based ODCNN.

The performance of the proposed model was evaluated and compared with that of other classifiers for K-fold values of 4,6,8, and 10 and 40%, 60% and 80% training data percentage. Highest values of accuracy, sensitivity, specificity, precision and F1-score for the compared approaches were achieved for K-fold value of 10 and 80% training data. The classification accuracy of the rapid navigation optimization-based ODCNN for the K-fold value of 10 was 97.26% with a performance improvement of 4.19% than the standard moth flame optimized- deep CNN. The sensitivity (recall) of the ODCNN also indicates an increase of 5.91%, with sensitivity value at 94.30% for the K-fold value of 10. The specificity of the rapid navigation optimized- deep CNN for a K-fold value of 10 was 99% with a performance improvement of 3.71% over the existing moth flame optimized - deep CNN. The precision of the rapid navigation optimized- deep CNN for a K-fold value of 10 was 90.30% with a performance improvement of 3.67% over the existing moth flame optimized - deep CNN. Table IV. presents a detailed comparative performance analysis of the various classifiers based on K- fold 10. A detailed comparative performance analysis of the various classifiers based on 80% training data is listed in Table V. The accuracy, sensitivity, specificity, precision and F1-score of the proposed rapid navigation optimization-based deep CNN classifier were 97.260%, 94.301%, 99%, 90.30% and 92.36%, respectively for the k-fold value 10 and 91.461%, 92.261%, 96.167%, 89.33% and 90.77%, respectively for 80% training data. The comparative analysis presented in Table IV and Table V clearly indicates that the proposed Rapid Navigation Optimization technique effectively tunes the 7-layered customized DCNN and enhances classification performance.

The accuracy, sensitivity (recall), specificity and precision of the rapid navigation optimization-based ODCNN and other classifiers for K-fold value of 10 are depicted in Fig. 7, 8, 9 and 10, respectively.

D. ROC Analysis for COVID-CT Dataset

The performance of the proposed rapid navigation optimization-based deep CNN depending on the false positive rates as well as the true positive rate for the for COVID-CT dataset is depicted in Fig. 11. When there is a minimal 10% of error, the sensitivity of the rapid navigation optimization-based deep CNN is 91.134% while the other existing moth flame-based deep CNN attains a sensitivity of 88.965%. When considering the minimal percentage of error as 90%, the sensitivity of the RNO - based deep CNN is improved to 97.164% which is 2.45% better than the existing moth flame-based deep CNN with a sensitivity of 94.788%.

E. Performance Analysis for COVID-CT Dataset with and without Lung Segmentation

Pre-preprocessing of chest CT images obtained either through hospitals or public benchmarking datasets is an essential step in design of machine learning algorithms. Segmentation of the lungs from the chest CT scans is a significant pre-processing step. Lung segmentation delineates the lung region (comprising of whole lung and the lung lobes) from the back ground in a chest CT image. In this research work, we applied region-based segmentation to extract the lung region from the chest CT image. We evaluated the impact of lung segmentation on the performance of the proposed model. It is evident that accuracy, precision, sensitivity (recall) and specificity have reduced by 20.35%, 19.34%, 22.35%, and 18.57% respectively, if CT scans are input without segmenting the Region-of Interest, that is the lung area. Fig. 12 depicts the comparative performance analysis of the proposed model with and without lung segmentation for K-Fold value 10. Table VI lists the performance analysis of the proposed model with and without lung segmentation for K-Fold value 10.

F. Performance Analysis for COVID-CT Dataset with and without Data Augmentation

Given the limited number of COVID-19 CT scans in the COVID-CT dataset, we performed data augmentation on the small dataset. The main objective of applying data augmentation was to evaluate our proposed model's performance and robustness on a larger and diverse dataset. We performed data augmentation by using MATLAB's inbuilt imageDataAugmenter function. Every CT image was subjected to rotation at various angles, and translation on the horizontal and vertical axes. The augmented dataset size is thrice as larger with variations of the data patterns. The proposed model exhibited a marginal increase in the performance metrics on the augmented dataset. Fig. 13 illustrates the performance analysis of the proposed model with and without data augmentation for K-Fold value 10. Table VII lists the performance analysis of the proposed model with and without data augmentation for K-Fold value 10.

G. Performance Analysis of Proposed RNO-optimized OD-CNN for COVIDx-CT-3A Dataset

To further validate and test performance of the proposed Rapid navigation optimization – based DCNN on a larger and different dataset, we utilized COVIDx-CT-3A. It is an open access dataset with 425,024 CT slices from 5,312 patients [29]. This section presents the performance analysis of RNO-optimized DCNN for COVIDx-CT-3A dataset. Evaluation was done for K-fold values of 4,6,8, and 10 and 40%, 60% and 80% training data. Table VIII reflects accuracy, precision, sensitivity (recall), specificity and F1-score for COVID-CT and COVIDx-CT-3A datasets for the k-fold value 10. Accuracy, precision, sensitivity, specificity, and F1-score for the k-fold value 10 were observed to be 89.35%, 85.08%, 90.49%, 89.57% and 87.70%, respectively.

Accuracy, precision, sensitivity, specificity, and F1-score for the COVIDx-CT-3A dataset were 88.46%, 87.62%, 86.55%, 87.12% and 87.08%, respectively for 80% training data. A comparative analysis for K fold 10 is shown in Fig. 14. Performance metrics for the two datasets with 80% training data are shown in Table IX. ROC curve for COVIDx-CT-3A dataset is illustrated in Figure 15.

H. Comparative Performance Analysis of Rapid Navigation Optimizer – based DCNN with Research Works using the COVID-CT Scan Dataset

The published research studies identified for a comparative analysis have used the same public dataset [10] as used in this proposed study. Table X presents a detailed comparative performance analysis of the proposed RNO - based deep CNN and research studies that have used the same dataset [10]. Performance analysis clearly indicates that our proposed model outperformed the other state-of-art methods. As COVID-19 lesions affect pixel intensity values in CT images, we exploited Haralick texture features to identify abnormalities. Feature selection through ANOVA and Chi-squared tests at the first stage assisted in identifying four relevant Haralick features and eliminating the correlated features.

To extract more complex and subtle features from the four texture features, contrast, correlation, entropy and sum entropy, we used a pre-trained ResNet101, followed by ReliefF-based feature selection. ReliefF algorithm considered 10 neighboring hits and misses to estimate the feature quality and relevance, thereby eliminating the redundant features and reducing feature vector size from 544x5001 to 1000x1.

The 1000 highly discriminating features were input to the 7-layer optimized DCNN and trained for only 30 epochs. The proposed RNO optimized 7-layer ODCNN is light-weight and has only 2k parameters. It exhibited high performance metrics to classify between COVID-19 and non-COVID -19 images.

Tuning of large number of parameters in DNNs is a critical issue. Our proposed optimizer combines the advantages of Moth-Flame optimizer and May fly optimizer to obtain accuracy of 97.26%, sensitivity of 94.30% and 99% specificity on a size-limited public dataset.

5. Conclusion and Future Work

Lack of availability of large repositories of annotated chest CT images was the primary motivation to develop a model that would learn efficiently from a size-limited dataset. Since the dataset is too small to train a DNN from scratch we exploited texture features of CT scan images, and rich feature representations from a pre-trained ResNet 101. We designed a robust joint feature descriptor that considers edge response values in different directions. As the joint

TABLE IV. Comparitive analysis of classifiers based on K-Fold 10 for COVID-CT dataset.

Classifiers	K-Fold 10				
	Acc(%)	Pre(%)	Sen(Recall)(%)	Spe (%)	F1 Score
K-Nearest Neighbor(KNN)	79.92	67.33	76.42	82.76	71.59
Support Vector Machine(SVM)	80.75	69.33	76.82	84.08	72.88
Feed Forward Neural Network	84.41	76.99	81.23	87.11	79.05
Pre-trained AlexNet Neural Network	87.57	77.92	84.34	92.71	81.00
ODCNN with Adam Optimizer	90.40	78.58	85.35	92.79	81.82
ODCNN with Mayfly Optimizer	90.55	83.22	87.68	94.86	85.39
ODCNN with Moth Flame Optimizer	93.18	86.63	88.72	95.33	87.66
ODCNN with proposed Rapid Navigation Optimizer	97.26	90.30	94.30	99.00	92.26

TABLE V. Comparative analysis of classifiers based on 80% training data for COVID-CT dataset.

Classifiers	Training Percentage 80%				
	Acc(%)	Pre(%)	Sen(Recall)(%)	Spe (%)	F1 Score
K-Nearest Neighbor(KNN)	78.24	65.91	74.56	81.08	69.96
Support Vector Machine(SVM)	78.93	68.06	75.21	83.23	71.46
Feed Forward Neural Network	82.24	72.89	79.51	85.39	76.06
Pre-trained AlexNet Neural Network	83.87	80.67	81.92	89.70	81.29
ODCNN with Adam Optimizer	86.59	83.30	83.26	91.85	83.28
ODCNN with Mayfly Optimizer	88.32	86.57	85.30	93.78	85.93
ODCNN with Moth Flame Optimizer	90.08	87.21	87.34	94.01	87.27
ODCNN with proposed Rapid Navigation Optimizer	91.46	89.33	92.26	96.16	90.77

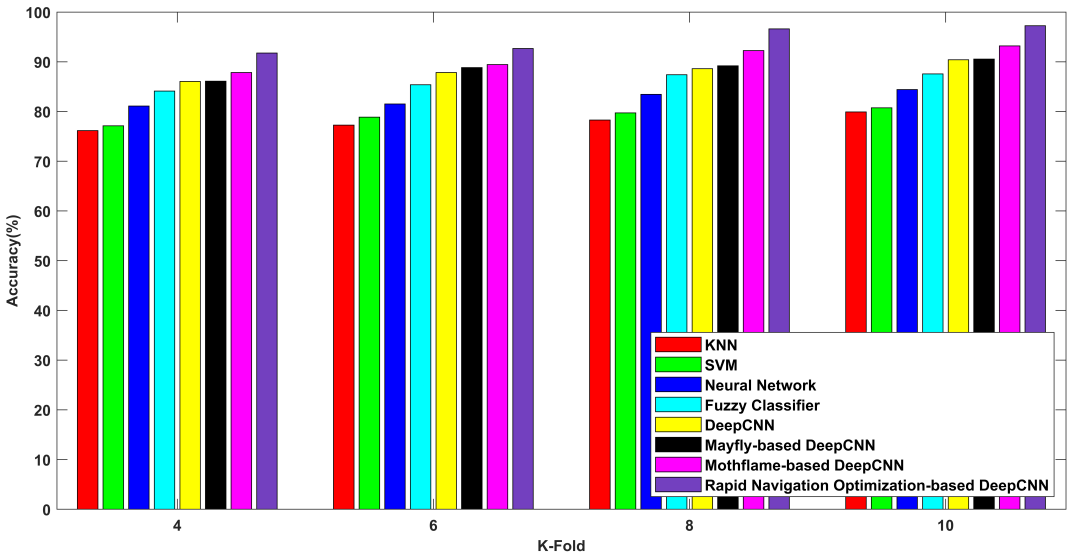

Figure 7. Comparative analysis of accuracy based on k-fold for COVID-CT dataset

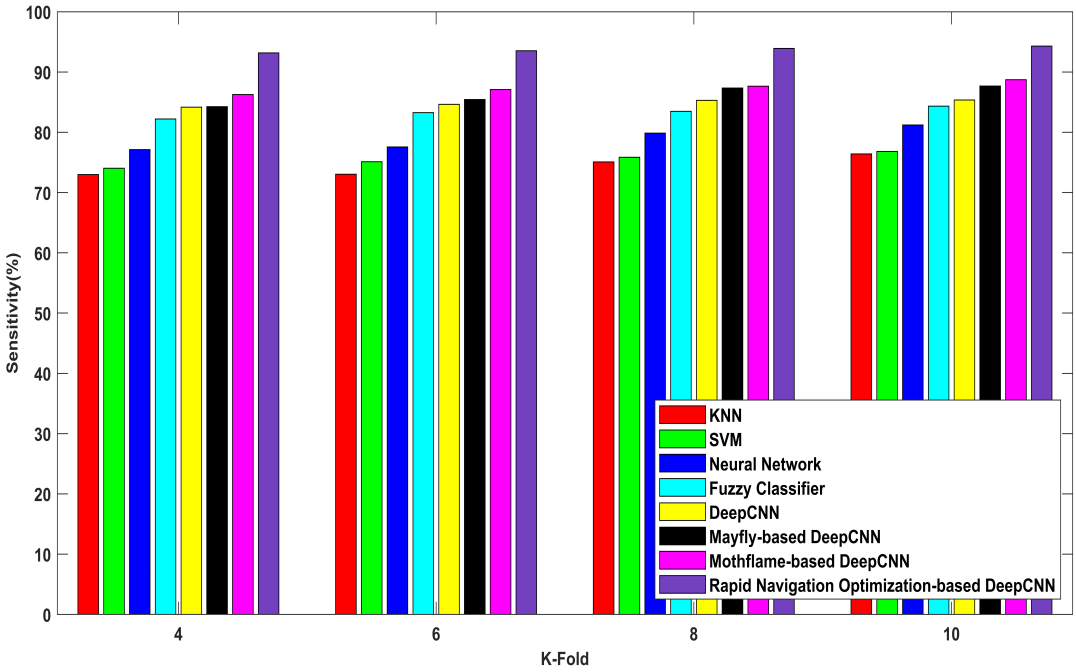

Figure 8. Comparative analysis of sensitivity (recall) based on k-fold for COVID-CT dataset

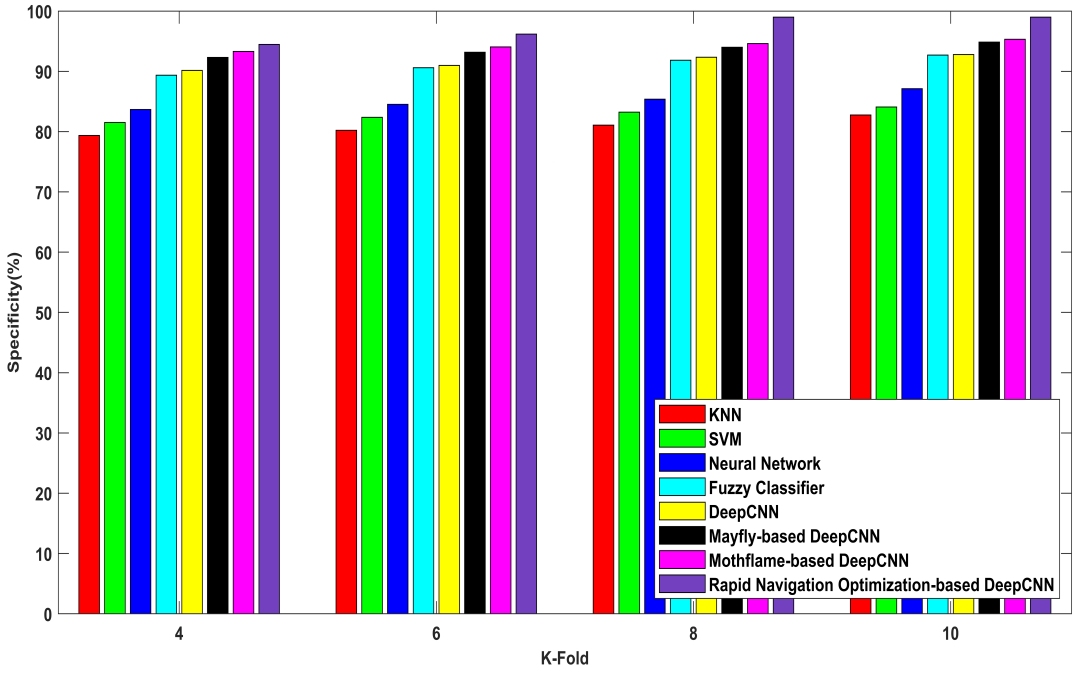

Figure 9. Comparative analysis of specificity based on k-fold for COVID-CT dataset

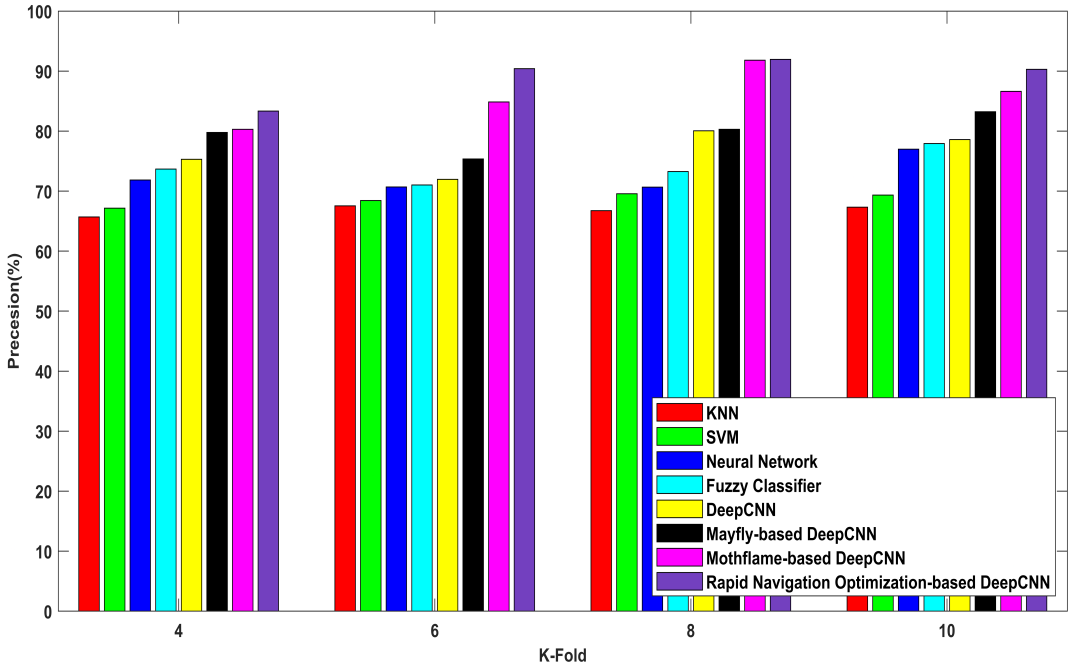

Figure 10. Comparative analysis of precision based on k-fold for COVID-CT dataset

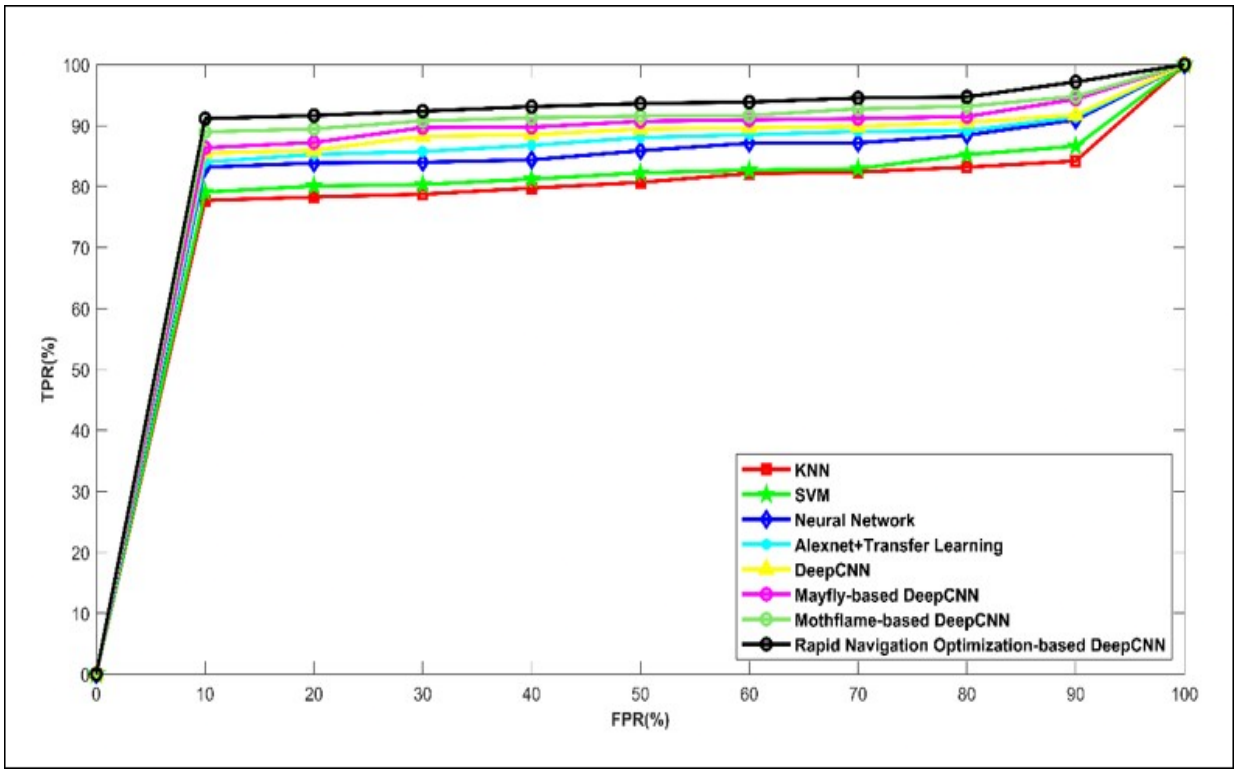

Figure 11. ROC analysis for COVID-CT dataset

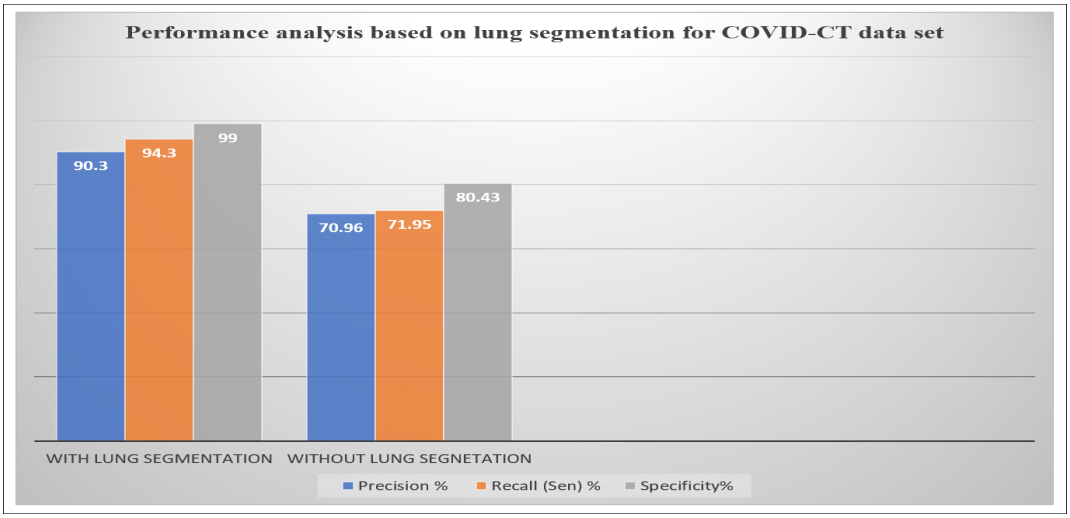

Figure 12. Performance analysis with and without lung segmentation

TABLE VI. Comparative analysis of RNO-Optimized DCNN based on lung segmentation for COVID-CT dataset.

Pre-processing	K-Fold 10				
	Acc(%)	Pre(%)	Sen(%)	Spe(%)	F1-score
Without lung segmentation	76.91	70.96	71.95	80.43	71.45
With lung segmentation	97.26	90.30	94.30	99.00	92.26

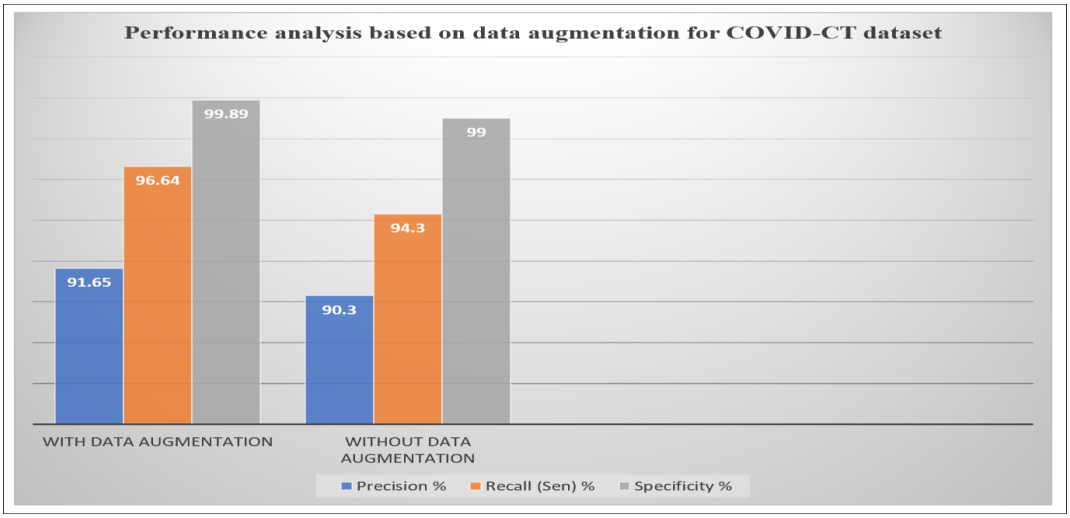

Figure 13. Performance analysis with and without data augmentation

TABLE VII. Comparative analysis of RNO-Optimized DCNN based on augmented COVID-CT dataset.

Pre-processing	K-Fold 10				
	Acc(%)	Pre(%)	Sen(%)	Spe(%)	F1-score
With augmentation	99.61	91.65	96.64	99.89	94.08
Without augmentation	97.26	90.30	94.30	99.00	92.26

TABLE VIII. Comparative analysis of RNO-Optimized DCNN based on K-Fold 10 for COVID-CT and COVIDx-CT-3A datasets.

Dataset	K-Fold 10				
	Acc(%)	Pre(%)	Sen(%)	Spe(%)	F1-score
COVID-CT dataset[10]	97.26	90.30	94.30	99.00	92.26
COVIDx-CT-3A dataset[29]	89.35	85.08	90.49	89.57	87.70

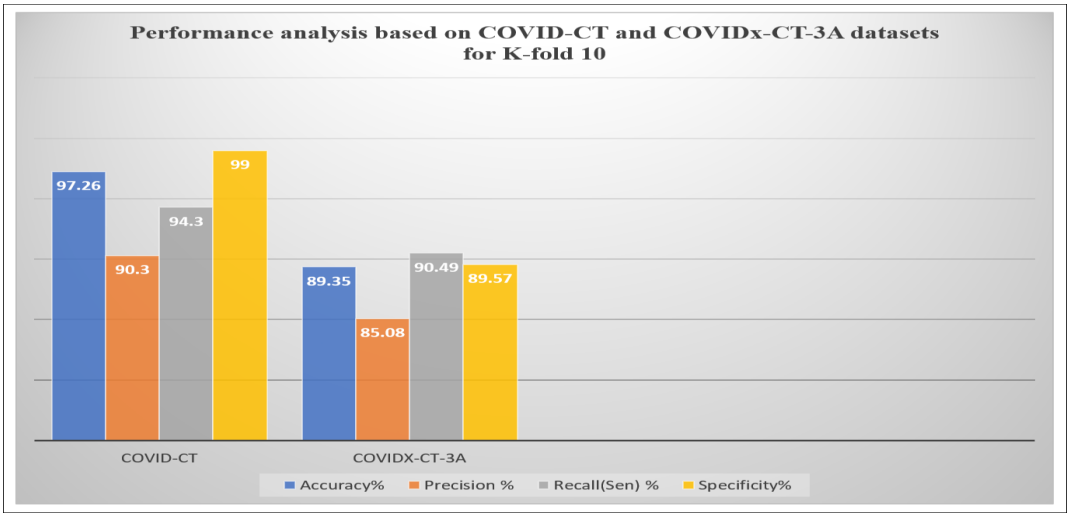

Figure 14. Comparative analysis for COVID-CT and COVIDx-CT-3A datasets

TABLE IX. Comparative analysis of RNO-Optimized DCNN based on 80% training data for COVID-CT and COVIDx-CT-3A datasets.

Dataset	Training Percentage 80%				
	Acc(%)	Pre(%)	Sen(%)	Spe(%)	F1-score
COVID-CT dataset[10]	91.46	89.33	92.26	96.16	90.77
COVIDx-CT-3A dataset[29]	88.46	87.62	86.55	87.12	87.08

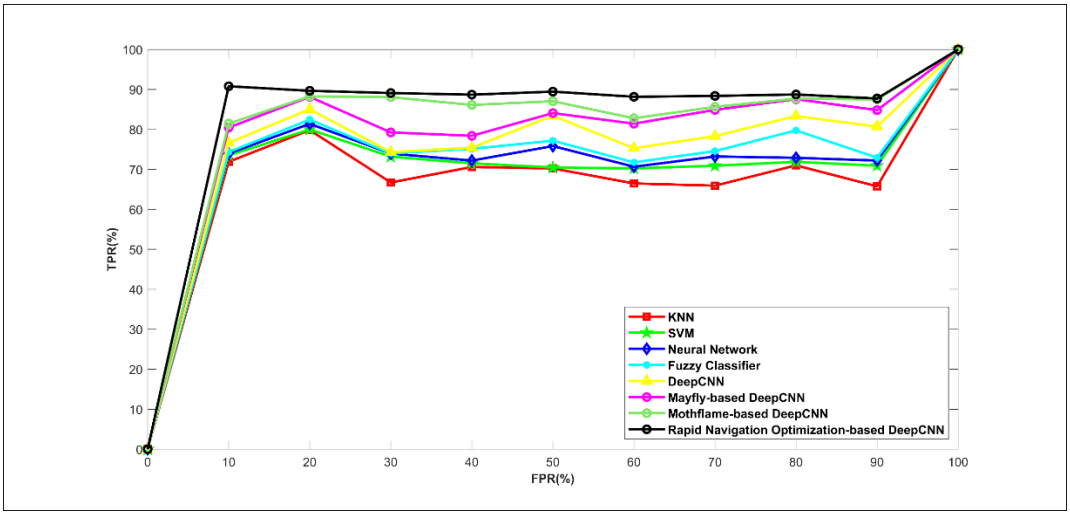

Figure 15. ROC analysis for COVIDx-CT-3A dataset.

feature descriptor does not rely solely on pixel intensities,

TABLE X. Comparative analysis of RNO with state-of-art.

S.No.	Reference	Acc(%)	Sen(%)	Spe(%)
1.	Punitha[9]	92.37	-	-
2.	Saha Sen[23]	90.00		
3.	Mishra[26]	88.34	88.13	90.51
4.	Shaban[27]	96	71	-
5.	**Proposed RNO-based ODCNN**	**97.26**	**94.30**	**99.00**

it is more robust against different variations and random noise. Incorporating the Directional Local Binary Pattern code (DLBP) eliminates the dependency of LDP code on the selection of 'k' most significant directional responses. Performance of the light-weight customized ODCNN was further enhanced by integrating two nature-inspired optimizers, viz. Moth-Flame Optimizer and Mayfly optimizer to tune the weights of the ODCNN. The performance of the proposed model was evaluated and compared with that of other classifiers for K-fold values of 4,6,8, and 10. The training data percentage was set at 40%, 60% and 80%. Highest values of accuracy, sensitivity and specificity for the compared approaches were achieved for K-fold value of 10 and 80% training data. The accuracy, sensitivity, and specificity of the proposed Rapid Navigation Optimization-based deep CNN classifier were 97.260%, 94.301%, 99% for the k-fold value 10, and 91.461%, 92.261%, and 96.167%, respectively for 80% training data. We evaluated our proposed model's performance and robustness on a larger and diverse dataset by augmenting the COVID-CT dataset. The model performed well on the augmented dataset, exhibiting a marginal increase in the performance metrics. To further validate and test performance of the proposed Rapid Navigation Optimization – based ODCNN, we evaluated the model's performance on a larger dataset, the COVIDx-CT-3A dataset. Accuracy, precision, sensitivity, specificity, and F1-score for the k-fold value 10 were observed to be 89.35%, 85.08%, 90.49%, 89.57% and 87.70%, respectively. Our light-weight model aimed at size-limited dataset has performed effectively on a larger and diverse dataset with substantial classification accuracies. Thus, we have verified the generalization capability of our proposed light-weight RNO-based ODCNN model. Performance analysis with related research works that have utilized COVID-CT dataset clearly indicates that our proposed model has outperformed the other state-of-the art methods. However, this research work is limited to detection of COVID-19 infection. Therefore, in the future the dataset shall be expanded to include CT images of other types of pulmonary infections, viz, pneumonia. The proposed model shall be modified appropriately to facilitate multi-class detection and classification.

Declaration

Conflict of Interest

The authors declare that there is no conflict of interest in this study.

Data Availability

Chest CT scans for this study were collected from a public dataset, which is available on https://github.com/UCSD-AI4H/COVID-CT. The second larger dataset, COVIDx-CT-3A was collected from a public dataset, which is available on https://kaggle.com/datasets/hgunraj/covidxct. This open access dataset was last updated in June 2022.

References

[1] Z. Yin, Z. Kang, D. Yang, S. Ding, H. Luo, and E. Xiao, "A comparison of clinical and chest ct findings in patients with influenza a (h1n1) virus infection and coronavirus disease (covid-19)," *American Journal of Roentgenology*, vol. 215, no. 5, pp. 1065–1071, 2020.

[2] D. Dong, Z. Tang, S. Wang, H. Hui, L. Gong, Y. Lu, Z. Xue, H. Liao, F. Chen, F. Yang *et al.*, "The role of imaging in the detection and management of covid-19: a review," *IEEE reviews in biomedical engineering*, vol. 14, pp. 16–29, 2020.

[3] J. Jin, D.-h. Gao, X. Mo, S.-p. Tan, Z.-x. Kou, Y.-b. Chen, J.-b. Cao, W.-j. Chen, Y.-m. Zhang, B.-q. Li *et al.*, "Analysis of 4 imaging features in patients with covid-19," *BMC Medical Imaging*, vol. 20, pp. 1–7, 2020.

[4] M. Chung, A. Bernheim, X. Mei, N. Zhang, M. Huang, X. Zeng, J. Cui, W. Xu, Y. Yang, Z. A. Fayad *et al.*, "Ct imaging features of 2019 novel coronavirus (2019-ncov)," *Radiology*, vol. 295, no. 1, pp. 202–207, 2020.

[5] S. T. H. Kieu, A. Bade, M. H. A. Hijazi, and H. Kolivand, "A survey of deep learning for lung disease detection on medical images: state-of-the-art, taxonomy, issues and future directions," *Journal of imaging*, vol. 6, no. 12, p. 131, 2020.

[6] P. Sawant and R. Sreemathy, "A survey on image processing and machine learning techniques for detection of pulmonary diseases based on ct images," in *Advanced Machine Intelligence and Signal Processing*. Springer, 2022, pp. 707–719.

[7] M. M. Hossain, M. A. A. Walid, S. S. Galib, M. M. Azad, W. Rahman, A. Shafi, and M. M. Rahman, "Covid-19 detection from chest ct images using optimized deep features and ensemble classification," *Systems and Soft Computing*, p. 200077, 2024.

[8] E. Hassan, M. Y. Shams, N. A. Hikal, and S. Elmougy, "Detecting covid-19 in chest ct images based on several pre-trained models," *Multimedia Tools and Applications*, pp. 1–21, 2024.

[9] S. Punitha, T. Stephan, R. Kannan, M. Mahmud, M. S. Kaiser, and S. B. Belhaouari, "Detecting covid-19 from lung computed tomography images: A swarm optimized artificial neural network approach," *IEEE Access*, vol. 11, pp. 12 378–12 393, 2023.

[10] X. Yang, X. He, J. Zhao, Y. Zhang, S. Zhang, and P. Xie, "Covid-ct-dataset: a ct scan dataset about covid-19," *arXiv preprint arXiv:2003.13865*, 2020.

[11] Z.-J. Gao, Y. He, and Y. Li, "A novel lightweight swin-unet network for semantic segmentation of covid-19 lesion in ct images," *Ieee Access*, vol. 11, pp. 950–962, 2022.

[12] I. Tuncer, P. D. Barua, S. Dogan, M. Baygin, T. Tuncer, R.-S. Tan, C. H. Yeong, and U. R. Acharya, "Swin-textural: A novel textural features-based image classification model for covid-19 detection on chest computed tomography," *Informatics in Medicine Unlocked*, vol. 36, p. 101158, 2023.

[13] J. Li, X. Luo, H. Ma, and W. Zhao, "A hybrid deep transfer learning model with kernel metric for covid-19 pneumonia classification using chest ct images," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 20, no. 4, pp. 2506–2517, 2022.

[14] M. S. Sadi, M. Alotaibi, P. Saha, F. Y. Nishat, J. Tasnim, T. Alhmiedat, H. Almoamari, and Z. Bassfar, "Cov-ctx: A deep learning approach to detect covid-19 from lung ct and x-ray images." *International Journal of Online & Biomedical Engineering*, vol. 19, no. 9, 2023.

[15] A. Farjana, F. T. Liza, M. Al Mamun, M. C. Das, and M. M. Hasan, "Sars covidaid: Automatic detection of sars cov-19 cases from ct scan images with pretrained transfer learning model (vgg19, resnet50 and densenet169) architecture," in *2023 International Conference on Smart Applications, Communications and Networking (SmartNets)*. IEEE, 2023, pp. 1–6.

[16] K. K. Mohbey, S. Sharma, S. Kumar, and M. Sharma, "Covid-19 identification and analysis using ct scan images: Deep transfer learning-based approach," in *Blockchain Applications for Healthcare Informatics*. Elsevier, 2022, pp. 447–470.

[17] M. R. Islam and M. Nahiduzzaman, "Complex features extraction with deep learning model for the detection of covid19 from ct scan images using ensemble based machine learning approach," *Expert Systems with Applications*, vol. 195, p. 116554, 2022.

[18] Q. Ren, B. Zhou, L. Tian, and W. Guo, "Detection of covid-19 with ct images using hybrid complex shearlet scattering networks," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 1, pp. 194–205, 2021.

[19] Y. Song, S. Zheng, L. Li, X. Zhang, X. Zhang, Z. Huang, J. Chen, R. Wang, H. Zhao, Y. Chong *et al.*, "Deep learning enables accurate diagnosis of novel coronavirus (covid-19) with ct images," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 18, no. 6, pp. 2775–2780, 2021.

[20] C. Li, L. Dong, Q. Dou, F. Lin, K. Zhang, Z. Feng, W. Si, X. Deng, Z. Deng, and P.-A. Heng, "Self-ensembling co-training framework for semi-supervised covid-19 ct segmentation," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 11, pp. 4140–4151, 2021.

[21] H. M. Balaha, E. M. El-Gendy, and M. M. Saafan, "Covh2sd: A covid-19 detection approach based on harris hawks optimization and stacked deep learning," *Expert systems with applications*, vol. 186, p. 115805, 2021.

[22] T. Kaur, T. K. Gandhi, and B. K. Panigrahi, "Automated diagnosis of covid-19 using deep features and parameter free bat optimization," *IEEE Journal of Translational Engineering in Health and Medicine*, vol. 9, pp. 1–9, 2021.

[23] S. Sen, S. Saha, S. Chatterjee, S. Mirjalili, and R. Sarkar, "A bi-stage feature selection approach for covid-19 prediction using chest ct images," *Applied Intelligence*, vol. 51, pp. 8985–9000, 2021.

[24] B. Wang, S. Jin, Q. Yan, H. Xu, C. Luo, L. Wei, W. Zhao, X. Hou, W. Ma, Z. Xu *et al.*, "Ai-assisted ct imaging analysis for covid-19 screening: Building and deploying a medical ai system," *Applied Soft Computing*, vol. 98, p. 106897, 2021.

[25] H. Yasar and M. Ceylan, "A novel comparative study for detection of covid-19 on ct lung images using texture analysis, machine learning, and deep learning methods," *Multimedia Tools and Applications*, vol. 80, pp. 5423–5447, 2021.

[26] A. K. Mishra, S. K. Das, P. Roy, S. Bandyopadhyay *et al.*, "Identifying covid19 from chest ct images: a deep convolutional neural networks based approach," *Journal of Healthcare Engineering*, vol. 2020, 2020.

[27] W. M. Shaban, A. H. Rabie, A. I. Saleh, and M. Abo-Elsoud, "A new covid-19 patients detection strategy (cpds) based on hybrid feature selection and enhanced knn classifier," *Knowledge-Based Systems*, vol. 205, p. 106270, 2020.

[28] P. Sawant and R. Sreemathy, "A review on texture feature analysis of chest computed tomography images for detection and classification of pulmonary diseases," in *International Conference on Communication and Intelligent Systems*. Springer, 2022, pp. 463–475.

[29] H. Gunraj, A. Sabri, D. Koff, and A. Wong, "Covid-net ct-2: Enhanced deep neural networks for detection of covid-19 from chest ct images through bigger, more diverse learning," *Frontiers in Medicine*, vol. 8, p. 729287, 2022.

[30] A. Humeau-Heurtier, "Texture feature extraction methods: A survey," *IEEE access*, vol. 7, pp. 8975–9000, 2019.

[31] A. M. Shabat and J.-R. Tapamo, "Directional local binary pattern for texture analysis," in *Image Analysis and Recognition: 13th International Conference, ICIAR 2016, in Memory of Mohamed Kamel, Póvoa de Varzim, Portugal, July 13-15, 2016, Proceedings 13*. Springer, 2016, pp. 226–233.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[33] R. J. Urbanowicz, M. Meeker, W. La Cava, R. S. Olson, and J. H. Moore, "Relief-based feature selection: Introduction and review," *Journal of biomedical informatics*, vol. 85, pp. 189–203, 2018.

[34] K. Zervoudakis and S. Tsafarakis, "A mayfly optimization algorithm," *Computers & Industrial Engineering*, vol. 145, p. 106559, 2020.

[35] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowledge-based systems*, vol. 89, pp. 228–249, 2015.

[36] L. Ma, C. Wang, N.-g. Xie, M. Shi, Y. Ye, and L. Wang, "Moth-flame optimization algorithm based on diversity and mutation strategy," *Applied Intelligence*, vol. 51, pp. 5836–5872, 2021.

Priya Sawant Assistant Professor, Marathwada Mitramandal's College of Engineering, Pune, MH, India and Research Scholar, department of Electronics and Telecommunication Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, India. Research areas include Signal Processing, Digital Image Processing, Machine Learning and Internet of Things.
Email: jopent8@gmail.com

R. Sreemathy Associate Professor, department of Electronics and Telecommunication Engineering, Pune Institute of Computer Technology, Savitribai Phule Pune University, India. Research areas include Signal Processing, Deep Learning, Machine Learning and Wireless Communication.
Email: rsreemathy@pict.edu